

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVA

DINO KAVGIĆ

ANDROID APLIKACIJA ZA POSREDOVANJE NEKRETNINAMA

ZAVRŠNI RAD

ČAKOVEC, 2017.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVA

DINO KAVGIĆ

ANDROID APP. FOR REAL ESTATEAPPLICATION

ZAVRŠNI RAD

Mentor: dr. sc. Bruno Trstenjak

ČAKOVEC, 2017.

Sažetak:

Cilj rada je razvoj aplikacije za posredovanje nekretninama koja će omogućiti izvršavanje aplikacije na web preglednicima i mobilnim uređajima. Aplikacija za posredovanje nekretninama namijenjena je za objavu oglasa o prodaji i kupovini nekretnina te pregled istih. Aplikacija je izrađena u ASP.NET tehnologiji. Aplikacija je responzivnog karaktera te podržava android uređaje upotrebom web view elemenata. U završnom radu izrađena je web aplikacija i mobilna aplikacija. Mobilna omogućuje prilagodbu asp.net aplikaciju na android uređaje. Web aplikacija i mobilna aplikacija sastoje se od modula za registraciju novih korisnika, modula za predaju oglasa, modula za prijavu registriranih korisnika, modula za pretraživanje oglasa prema postavljenim uvjetima. Da se osigura prilagodljivost aplikacije na sve rezolucije zaslona korištena je Bootstrap biblioteka čija su svojstva opisana u radu. U aplikaciji je implementirana Google kartu na kojoj se prikazuje lokacija nekretnine te modula administratora koji upravlja oglasima. Aplikacija za zapisivanje podataka koristi MySQL bazu podataka. Također opisane su metode koje se koriste za manipulaciju i dohvaćanje traženih podataka te je detaljno opisano njihovo pozivanje i realizacija. Provedeno je testiranje aplikacija, a rezultati i sam postupci opisani su u poglavlju 6.

Ključne riječi: android aplikacija, nekretnine, .NET, MySQL.

Sadržaj:

1. Uvod.....	1
2. Cilj završnog rada i doprinos	2
3. Platforme za izradu aplikacije.....	3
3.1 Android operacijski sustav	3
3.1.1 Arhitektura Android sustava	3
3.2 ASP.NET tehnologija	7
3.3 Bootstrap tehnologija.....	9
3.4 Entity Framework	10
3.5 jQuery	10
4. Struktura aplikacije	11
4.1 Struktura web aplikacije	11
4.2 Struktura mobilne aplikacije.....	13
4.2.1 Implementacija aplikacije	14
4.3 Baza podataka.....	17
4.3.1 Odnos relacija baze podataka.....	18
4.3.2 Opis strukture tablica baze podataka	18
4.3.3 Povezivanje baze sa aplikacijom	18
5. Funkcionalnost komponenti aplikacije	19
5.1 Modul za prijavu korisnika.....	19
5.2 Modul za registraciju korisnika	21
5.3 Administracija podataka i korisnika	23
5.4 Modul za predaju oglasa.....	26
5.5 Modul za pretraživanje oglasa	28
6. Konfiguriranje aplikacija	34
7. Testiranje aplikacije	35
8. Zaključak.....	38

1. Uvod

Razvoj informacijsko-telekomunikacijske tehnologije doveo je do toga da se sve više koriste napredna računala koja imaju napredne računalne i komunikacijske sposobnosti. Ova pojava omogućila je da se poslovi koji su se standardno obavljali pomoću računala sada obavljaju putem mobilnih uređaja. IDC¹ statistički podaci za drugi kvartal 2015. godine govore da je prodano 341.5 milijuna smartphone uređaja. Rast prodaje mobilnih uređaja tijekom godine dosegnuo je 13% udjela na globalnom svjetskom tržištu mobilne industrije[1]. Vodeći na tržištu i dalje su Android i iOS platforme. Android drži 82,8% tržišnog udjela, a iOS 13,9%[1]. Ovi podaci nedvojbeno ukazuju na sve veću važnost posjedovanja vlastite mobilne aplikacije kao alata za promicanje vlastitog poslovanja. Iz tog razloga mobilna aplikacija za posredovanje nekretninama može biti jako korisna u svakodnevnom životu. Tržištu nekretnina može pristupiti putem mobilnog uređaja. Aplikacija omogućuje korisnicima pretraživanje te jednostavnu i brzu objavu oglasa podataka o nekretninama. Kada korisnik otvori listu objavljenih nekretnina svoju odluku o željenoj nekretnini može donijeti pregledavajući detaljne informacije o nekretninama uključujući cijenu, slike, kvadraturu i slično. U završnom radu prikazan je proces razvoja web i mobilne aplikacije, objašnjene su korištene tehnologije te je prikazana struktura baze koja je korištena na razini web poslužitelja. Aplikacija omogućuje pohranjivanje podataka i slika na web poslužitelj. U aplikaciji su definirane dvije vrste korisnika. Autorizirani korisnici aplikacije mogu uređivati sadržaj oglasa koji se objavljuje. Ostali korisnici, koji nisu autorizirani, mogu samo pretraživati objavljene sadržaje. Aplikacija u svojoj strukturi ima ugrađene određene filtere za pretraživanje objavljenih oglasa nekretnina prema nekim osnovnim svojstvima tražene nekretnine.

¹IDC - svjetska agencija za istraživanje ICT tržišta

2. Cilj završnog rada i doprinos

Cilj ovog završnog rada je izrada aplikacije za posredovanje nekretninama koja se može izvršavati na internet preglednicima i mobilnim uređajima koji imaju android podršku. Cilj izrade je ponuditi mogućnosti odabira nekretnine prema željenim parametrima i olakšati zainteresiranim korisnicima kupnju nekretnine te pristup pregleda željenih oglasa koji drugi korisnici oglašavaju.

Aplikacija je osmišljena da korisniku omogući jednostavnost u korištenju. Tako da korisnici koji imaju osnovna znanja informatičke pismenosti, jer aplikacija sama „vodi“ korisnika do željenog cilja i daje eventualna upozorenja ako korisnik pogriješi kod unosa podataka, izgubi internet vezu i sl.

Aplikaciju mogu koristiti poduzeća, dionička društva, ali i ostali korisnici koji se žele oglasiti i olakšati prodaju svoje nekretnine. U početku izrade ovog projekta, a kod realizacije ideje ovog projekta, aplikacija je trebala biti ograničena samo za određeno poduzeće. Odlučeno je da aplikacija bude univerzalnog karaktera što će rezultirati njenom većom iskorištenosti. Aplikaciju je moguće nadograđivati te ju učiniti kompleksnijom. Ovo je moj prvi ozbiljan projekt i za mene predstavlja novo iskustvo jer sam se prvi puta susreo sa pojedinim alatima za izradu ovog projekta koji su upotrijebili dizajn i učinili ovu aplikaciju modernom i jednostavnom za korištenje.

3. Platforme za izradu aplikacije

U ovom poglavlju ukratko su opisane platforme korištene u izradi ovog projekta.

3.1 Android operacijski sustav

Android je operacijski sustav za mobilne uređaje koji koristi Linux kernel. Projekt Android započeo je 2005. godine kada je Google kupio tvrtku naziva Android Inc. koju su osnovali Andy Rubin, Rich Miner, Nick Sears i Chris White u listopadu 2003. godine. Google je pokretač i glavni začetnik Androida. Barem načelno za razvoj Androida je zadužen Open HandsetAlliance². Androidov kod je potpuno otvoren od 2008. godine i objavljen je pod Apache licencom. Iako je Android slobodan operacijski sustav, SDK (engl. *software development softver*, softver za izradu aplikacija za ovaj OS) nije u potpunosti otvoren[2]. Posljednja stabilna verzija androida je Android 7.0 "Nougat" razvijena 22. kolovoza 2016. godine. Podržane platforme za tu inačicu su 32-bitni i 64-bitni ARM, x86, MIPS, MIPS64 te se temelji na linuxkernelu[3].

3.1.1 Arhitektura Android sustava

Temelj Android platforme je Linux kernel. Korištenje Linux kernela Androidu omogućuje korištenje ključnih sigurnosnih značajki i omogućuje proizvođačima uređaja razvoj hardverske upravljačke programe za dobro poznati kernel. U sljedećem dijelu rada pikazane su glavne komponente android platforme koje se mogu vidjeti na slici 1.

²OpenHandsetAlliance- udruga proizvođača elektronike.

Sloj hardverske apstrakcije (HAL)

Sloj hardverske apstrakcije (HAL) (engl. Hardware AbstractionLayer) pruža standardna sučelja koja izlažu hardverske mogućnosti uređaja na višu razinu Java API okvira. HAL se sastoji od više knjižničnih modula od kojih svaka biblioteka implementira sučelje za određenu vrstu hardverske komponente kao što je fotoaparati ili Bluetooth modul. Kada okvir API šalje zahtjev za pristup hardveru uređaja, sustav Android učitava modul za biblioteku te hardverske komponente.

Android runtime

Android runtime je sloj koji služi za pokretanje aplikacija. Sastoji se od dvije važne komponente. Prva su takozvane "Core libraries" odnosno knjižnice koje sadrže većinu knjižnica programskog jezika Java. Druga komponenta je DalvikVirtualMachine koji pokreće aplikacije kao zasebne procese odnosno kao instance virtualnog stroja. DVM pretvara Java class datoteke u svoj vlastiti format (.dex), kako bi bile optimizirane za minimalni utrošak memorije. Neke od glavnih značajki ART-a (Android runtime).

uključuju sljedeće:

- Optimizirano sakupljanje smeća (engl. garbagecollection)
- Bolja podrška za uklanjanje pogrešaka
- Bolja podrška za uklanjanje pogrešaka, izvješćivanje o padu aplikacija te mogućnost postavljanja nadzornih točaka za praćenje određenih polja.

³OpenGL ili javna grafička biblioteka (eng. *Open graphics library*) - grafički 3d standard stvoren od strane silicon graphics kompanije.

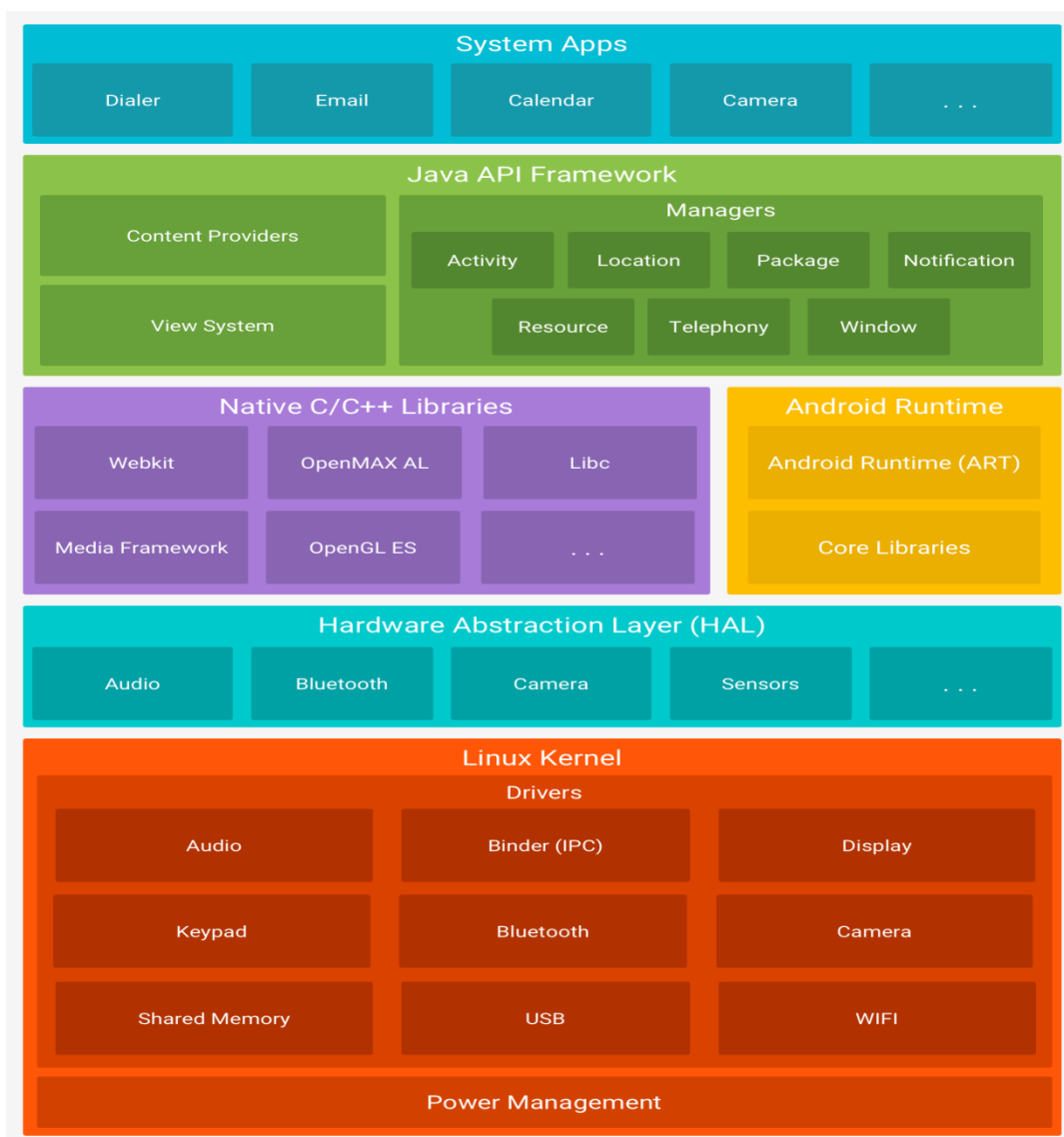
Izvorne C / C++ knjižnice

Mnoge komponente koje se nalaze u jezgri su slojevi sustava Android, kao što su ART i HAL, izgrađene su od biblioteka koje se sastoje od izvornog koda napisane u C i C++ jeziku. Android platforma omogućuje da aplikacije koje su pisane u java prog. jeziku mogu koristiti razne funkcije koje su implementirane u nekim od tih izvornih knjižnica. Na primjer, pristup OpenGL³ ostvaruje kroz Java OpenGL API Android okvir. Na taj način ostvarena je podrška za crtanje i manipuliranje 2D i 3D grafike u aplikaciji.

Java APIframework

Cijeli Android operativni sustav (engl. OS) skup značajki dostupan je putem API-ja napisanih u Java jeziku. Ovi API-i oblikuju programske blokove za stvaranje android aplikacija. Na taj način pojednostavljena je ponovna uporaba osnovnih, modularnih komponenti sustava i usluga, a omogućuje sljedeće:

- Bogati i proširivi sustav elemenata koji se mogu koristiti za izradu korisničkog sučelja aplikacije uključujući popise, rešetke, tekstne okvire, gumbe pa čak i ugrađeni web preglednik,
- Upravitelj resursa koji omogućuje pristup resursima koji nisu kodovi kao što su lokalizirani nizovi, grafike i datoteke izgleda,
- Upravitelj obavijesti koji svim aplikacijama omogućuje prikaz prilagođenih upozorenja na traci statusa,
- Upravitelj aktivnosti koji upravlja životnim ciklusom aplikacija i pruža zajedničku navigaciju unatrag,
- Davatelji sadržaja koji omogućuju aplikacijama pristup podacima iz drugih aplikacija kao što je aplikacija Kontakti ili za dijeljenje vlastitih podataka,
- Razvojni programeri imaju puni pristup istim okvirnim API-a koje koriste aplikacije sustava Android.



Slika 1. Android platforma

Izvor: <https://developer.android.com/guide/platform/index.html>

3.2 ASP.NET tehnologija

Prednost ove tehnologije je što su web aplikacije pisane na sličan način kao i desktop aplikacije uz poštivanje pravila objektno orijentiranog programiranja. Najvažnija razlika u odnosu na ASP je to što se sada kod prevodi i izvodi unutar CLR-a (engl. Commonlanguage runtime). Sam kod moguće je pisati u bilo kojem programskom jeziku za koji postoji prevodilac. Jedna od velikih novosti su i web servisi. ASP.NET-a povezan je sa .NETframeworkom. .NETframework je ogroman, hijerarhijski uređen skup klasa, struktura, sučelja i ostalih programskih elemenata. Velika prednost ASP.NET-a u odnosu na neka druga rješenja je brzina. Ta je brzina direktna posljedica toga što se ASP.NET aplikacije prevode. Prevođenje se odvija u dva stupnja. U prvom stupnju kod se prevodi u "međujezik" nazvan MSIL (engl. Microsoft Intermediate Language). Drugi stupanj prevođenja događa se u trenutku kada se aplikacija izvršava.

Jedan od najvažnijih aspekata ASP.NET-a je to što se izvršava unutar CLR okruženja. Uloga CLR-a je automatsko upravljanje memorijom: "skupljanje smeća", pružanje podataka o dostupnim klasama, njihovim članovima, tipovima podataka i ostalim programskim elementima te obrađivanje pogrešaka, višedretvenost i slično. Za razliku od ASP-a čiji objektni model predstavlja samo tanki sloj iznad HTTP-a i HTML-a, ASP.NET nudi pravi objektno orijentirani model. Najbolji primjer za to su serverske kontrole kojima je moguće manipulirati preko implementiranih metoda dok su detalji vezani za HTML kod skriveni od strane programera.

Jedan od osnovnih dijelova ASP.NET aplikacije su web forme čija je uloga vezana uz prikaz same stranice prema korisniku. U web formama radi se o objektno orijentiranom pristupu, a gdje se nakon učitavanja dokumenta generiraju odgovarajući objekti i događaji.

Također, jedan od važnijih dijelova ASP.NET-a su serverske kontrole. Tu se radi o klasama .NETframeworka koje predstavljaju vizualne elemente na formi. ASP.NET nudi podršku za veliki broj kontrola koje možemo svrstati u sljedeće kategorije:

1. HTML serverske kontrole su kontrole u koju skupinu pripadaju obične HTML

kontrole deklarirane sa atributom `runat="server"`.

2. Web kontrole su kontrole koje uz standardnu klasu HTML-a sadrže određena svojstva i metode koje programerima omogućuju lakši pristup i manipulaciju. U ovu skupinu pripadaju između ostalih HyperLink, ListBox te Button kontrole.
3. Rich kontrole - ovakve kontrole imaju mogućnost generiranja većih dijelova HTML i JavaScript koda kojim prikazuju nešto kompleksnije dijelove korisničkog sučelja. Primjer kontrola iz ove skupine su Calendar, AdRotator te TreeView kontrola.
4. Validacijske kontrole - ove kontrole zadužene su za validaciju vrijednosti koje korisnik unosi preko forme.
5. Podatkovne kontrole - radi se o kontrolama namijenjenim ispisu i editiranju većih količina podataka.
6. Navigacijske kontrole koriste se za navigaciju unutar hijerarhije web aplikacije.
7. Login kontrole - koriste se umjesto ručnog pisanja koda koji bi se bavio autentikacijom korisnika, praćenjem njihovog statusa, registracijom i sličnim problemima ASP.NET nudi već gotove kontrole za to.
8. Web parts kontrole - u ovu skupinu pripadaju kontrole koje nude podršku za izgradnju modularnih web portala.
9. ASP.NET mobilne kontrole - radi se o skupu kontrola koje nude podršku za mobilne klijente poput ručnih računala i mobilnih telefona.
10. Za definiranje izgleda dijelova web stranice koriste se CSS teme i masterpage. Pomoću CSS-a definira se izgled stranice, ali je izgled stranice ograničen na određene attribute kojima je moguće formatirati stranicu (na primjer font, margine, pozadina, boje i slično). Teme omogućuju definiranje jer pomoću njih možemo definirati stilove i attribute bilo koje kontrole. Razlika između tema i CSS-a je u tome što je podrška za CSS implementirana unutra samog preglednika dok su teme implementirane unutar .NETframeworka na samom poslužitelju.

11. Teme i CSS nam omogućuju definiranje izgleda dijelova web stranice, ali njima nije moguće odrediti strukturu i raspored elemenata. Za rješenje ovog problema koristi se masterpages. To su zapravo predlošci stranica pomoću kojih je moguće definirati fiksni sadržaj te prostor za umetanje ostatka sadržaja. Prednost ovakvog pristupa je i to što se u slučaju izmjene predloška automatski mijenja izgled svih stranica[3].

3.3 Bootstrap tehnologija

Bootstrap je besplatan opensourceframework za razvoj dizajna web stranica i web aplikacija. Bootstrapframework temelji se na HTML, CSS i JavaScript (JS) skripti. On dinamički upravlja izgledom web stranica što omogućuje da su one responsivne. Mark Otto i Jacob Thornton razvili su Bootstrap na Twitteru. Softver je nekad bio poznat kao Twitter Blueprint i ponekad se naziva Twitter Bootstrap[6].

Prednosti Bootstrapa su te što ima ugrađen set alata i biblioteka za kreiranje fleksibilnih i responsive web formi sa svim pripadajućim elementima. Laka i jednostavna izrada web forme i interfejsa koja je prilagodljiva svim veličinama ekrana.

Nedostaci su što aplikacija ne daje aplikaciji specifičan izgled nego sve aplikacije imaju sličnu web formu [7].

Bootstrap se povezuje s aplikacijom koja se koristi tako da se u <head>tag doda link na njegov .css file:

Kod 1. Povezivanje s Bootstrap-ovim css i javascript datotekama

```
<link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstr
ap.min.css"/>
<scriptsrc="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bo
otstrap.min.js"></script>
```

3.4 Entity Framework

Entity Framework skup tehnologija u ADO.NET-u koji podržavaju razvoj softverskih aplikacija orijentiranih na podatke omogućuje razvojnim programerima rad s podacima u obliku objekata i svojstava specifičnih za određenu domenu kao što su korisnici i korisničke adrese, a bez potrebe da se bave tablicama i stupcima baze podataka u kojima se ti podaci pohranjuju. Pomoću EntityFrameworka razvojni programeri mogu raditi na višoj razini apstrakcije kada se bave podacima te mogu stvoriti i održavati aplikacije usmjerene na podatke s manje kodova nego u tradicionalnim aplikacijama. Budući da je entitetski okvir sastavni dio .NET okvira, Entity Framework aplikacije mogu se izvoditi na bilo kojem računalu na kojem je instaliran .NET Framework koji počinje s verzijom 3.5 SP1. Pomoću njega eliminira se stvaranje većine programskog koda za pristup podacima[8].

3.5 jQuery

jQuery je JavaScript biblioteka otvorenog koda koja omogućuje interakcije između HTML dokumenta ili preciznije, objektnog modela dokumenta (DOM) i JavaScripta. Specifičnosti biblioteke jQuery:

- jQuery biblioteka otvorenog je koda i riječ je o projektu sa licencama MIT i GNU General PublicLicense (GPL).
- Biblioteka je veličine (18 kB u minizimiranom obliku) i gzip komprimirana (nekomprimirana je 114 kB).
- Biblioteka ima veliki fond programskih modula koji se postupno širi otkad se biblioteka JQuery pojavila.
- Njen API je potpuno dokumentiran uključujući primjere ugrađenog koda.
- Namjenski je napravljena tako da se izbjegnu konflikti s drugim JavaScript bibliotekama.
- Razvija se otvoreno što znači da svatko može doprinijeti ispravljanju grešaka, napretku i razvoju.

- Usvaja specifikacije organizacije W3C preko web preglednika. Evo primjera: jQuery podržava veliki dio CSS3 selektora.
- Trenutno se testira i optimizira za razvoj na modernim preglednicima (Chrome 1, ChromeNightly, IE 6, IE 7, IE 8, Opera 9.6, Safari 3.2, WebKitNightly, Firefox 2, Firefox 3, Firefox Nightly).
- Njezina dokumentacija opisuje elemente (npr. API čitač, aplikacije podsjetnike) uključujući i API čitač van mreže (AIR aplikacija).
- JQuery se nadograđuje na koncepte koje većina projekatara i dizajnera već razumije (na primjer, CSS i HTML)[9].

4. Struktura aplikacije

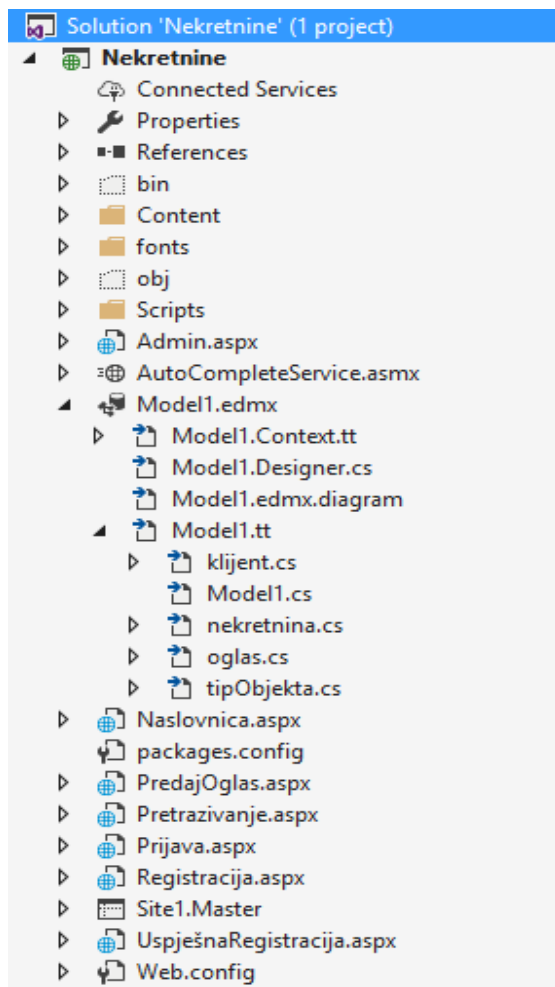
Aplikacija za posredovanje nekretninama sastoji se od mobilne aplikacije i web aplikacije. Rješenje projektnog zadatka osmišljeno je tako da korisnici mogu pristupati aplikaciji putem mobilnog uređaja. Aplikacija se pokreće preko android sustava koji pomoću web view elemenata pristupa web aplikaciji. Za razvoj aplikacije korištene su ASP.NET i android platforme. Aplikacija se sastoji od modula za pretraživanje, modula za registraciju, modula za predaju oglasa, modula za prijavu te modula administratora kojem se pristupa pomoću određenog linka o kojem će se dijelom kasnije govoriti.

4.1 Struktura web aplikacije

Ovo poglavlje opisuje module koje sadrži aplikacija, njihovu međusobnu povezanost te koje se informacije međusobno šalju. Aplikacija se sastoji od modula:

- Modul za prijavu korisnika - u navedenom modulu korisnik izvršava prijavu sa svojim podacima koje je naveo u registraciji korisnika.
- Modul za registraciju korisnika - u navedenom modulu korisnik upisuje svoje osobne podatke, a pomoću kojih se prijavljuje u aplikaciju.
- Administracija podataka - u navedenom modulu izvršavaju se administrativne radnje oglasima (upravljanje oglasima).

- Modul za predaju oglasa - u navedenom modulu korisnik pohranjuje svoj oglas sa podacima o nekretnini u bazu oglasa.
- Modul za pretraživanje oglasa - u navedenom modulu korisnici unose parametre pretrage prema kojima se filtriraju rezultati. Korisnici također mogu izvršiti pregled svih oglasa bez upisivanja podataka.



Slika 2. Mapa projekta

Izvor: Vlastiti izvor

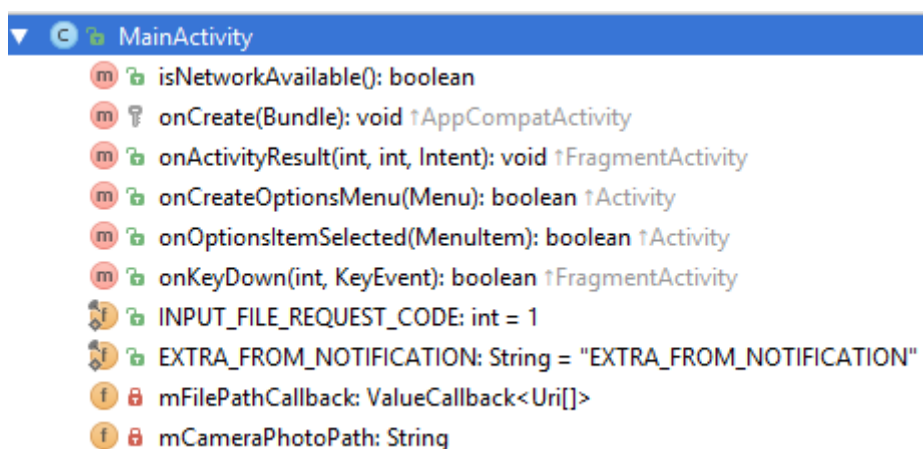
Slika 2. prikazuje organizaciju direktorija unutar projekta web aplikacije. U vršnom direktoriju nalaze se web forme, model baze podataka te web.config datoteka koja je

XML datoteka. Direktorij Content sadrži slike te CSS datoteke. Direktorij fonts sadrži fontove korištene za ikone te direktorij Scripts sadrži javascript knjižice i datoteke.

4.2 Struktura mobilne aplikacije

Struktura mobilne aplikacije sastoji se od jedne aktivnosti 'MainActivity', prikazane na slici 3. koja sadrži sve funkcionalnosti aplikacije. Aktivnost predstavlja jedan ekran sa korisničkim sučeljem čije se stanje prije prikazivanju korisniku inicijalizira, privremenim izlaskom iz aplikacije stanje se može pauzirati, a potpunim gašenjem aplikacije se uništava.

Unutar aktivnosti 'MainActivity' prikazuje se web stranica koja se nalazi na serveru. U slučaju da mobilni uređaj nema internet konekciju, prikazuje se posebna stranica 'error.html' koja sadrži odgovarajuću poruku.



Slika 3. Sadržaj aktivnosti 'MainActivity'

Izvor: Vlastiti izvor

4.2.1 Implementacija aplikacije

Jedna od važnijih metoda koja se koristi u aplikaciji je metoda `OnCreate` u kojoj se inicijalizira glavna aktivnost `MainActivity`. Ovdje se inicijalizira `WebView` element pomoću kojega web stranicu prikazujemo unutar android aplikacije. `WebView` omogućuje prikaz web stranica unutar android aplikacije. Kod 2. prikazuje postavljanje URL-a web stranice u `WebView` objekt kao i ostalih postavaka kao što su omogućavanje izvršavanje javascript koda, postavljanje pozadinske boje. Budući da aplikacija treba prikazivati vanjsku web stranicu unutar aplikacije, a ne u nekom od internet preglednika tog mobilnog uređaja, postavlja se i `WebViewClient` koji omogućava takvo ponašanje.

Kod 2. Inicijalizacija `WebView`-a

```
// url na kojoj se nalazi web aplikacija
Stringurl = "http://lianleli.gear.host/";
webview= (WebView) findViewById(R.id.webView);

// omogućavanje javascripta unutar webview-a
webview.getSettings().setJavaScriptEnabled(true);
webview.getSettings().setCacheMode(WebSettings.LOAD_NO_CACHE);
webview.invalidate();
webview.setBackgroundColor(Color.TRANSPARENT);
webview.setBackgroundResource(R.drawable.background);

// postavljanje webview klijenta
webview.setWebViewClient(newWebViewClient() {
    publicbooleanshouldOverrideUrlLoading(WebViewview, Stringurl)
    {
        returnsuper.shouldOverrideUrlLoading(view, url);
    }
});
```

Kod 3. prikazuje izvorni kod koji pomoću metode `isNetworkAvailable` provjerava postojanje veze prema internetu. Ukoliko postoji izvršava se učitavanje web aplikacije sa url-a. Ako ne postoji u `webview` se učitava `error.html` stranica te se poziva `alertDialog` unutar kojeg se prikaže poruka o nedostupnosti internetske veze. Na klik gumba OK postavlja se događaj koji će pozvati metodu koja će izaći iz aplikacije.

Kod 3. Provjera internet veze i učitavanje aplikacije

```
if (isNetworkAvailable()){
    // učitavanje web stranice u webview
    webView.loadUrl(url);
}
else{
    // učitavanje error stranice u webview
    webView.loadUrl("file:///android_asset/error.html");
    alertDialog.setTitle("Greška");
    alertDialog.setMessage("Internet veza nije dostupna.");
    alertDialog.setButton("OK", new
        // definiranje izlaza iz aplikacije na klik ekrana
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                MainActivity.this.finish();
                System.exit(0);
            }
        });
    alertDialog.show();
}
```

Za odabir slike pri unošenju oglasa koristi se metoda `OnShowFileChooser` koja odašilje `Intent` koji će korisniku omogućiti odabrati željenu sliku s lokalne memorije mobilnog uređaja.

Kod 4. prikazuje isječak koda koji definira `Intent` koji se odašilje operacijskom sustavu. Definira se akcija i kategorija `Intenta` te se postavlja vrsta datoteke za odabir.

Kod 4. Definiranje `Intenta` korištenog za odabir slike s memorije mobilnog uređaja

```
Intent contentSelectionIntent =
    new Intent(Intent.ACTION_GET_CONTENT);
contentSelectionIntent.addCategory(Intent.CATEGORY_OPENABLE);

// definiranje vrste sadržaja koji se odabire
contentSelectionIntent.setType("image/*");
```

Nakon odabira slike poziva se `onActivityResult` metoda koja kao parametar prima `Intent` koji sadržava `URI` (putanju) do slike na mobilnom uređaju.

Kod 5. Postavljanje putanje do slike koju je korisnik odabrao na mobilnom uređaju

```
public void onActivityResult (int requestCode, int resultCode,
Intent data) {
if(requestCode != INPUT_FILE_REQUEST_CODE
|| mFilePathCallback == null) {
super.onActivityResult(requestCode, resultCode, data);
// pri neuspješnom odabiru slike napusti funkciju
return;
}
Uri[] results = null;
if(resultCode == Activity.RESULT_OK) {
if(data != null) {

                // ako je odabir slike uspješan, rezultat se parsira
                // u listu URI-ja jer može sadržavati više slika
String dataString = data.getDataString();
if (dataString != null) {
results = new Uri[]{Uri.parse(dataString)};
}

                // postavljanje putanje do slike
mFilePathCallback.onReceiveValue(results);
mFilePathCallback = null;
return;
}
```

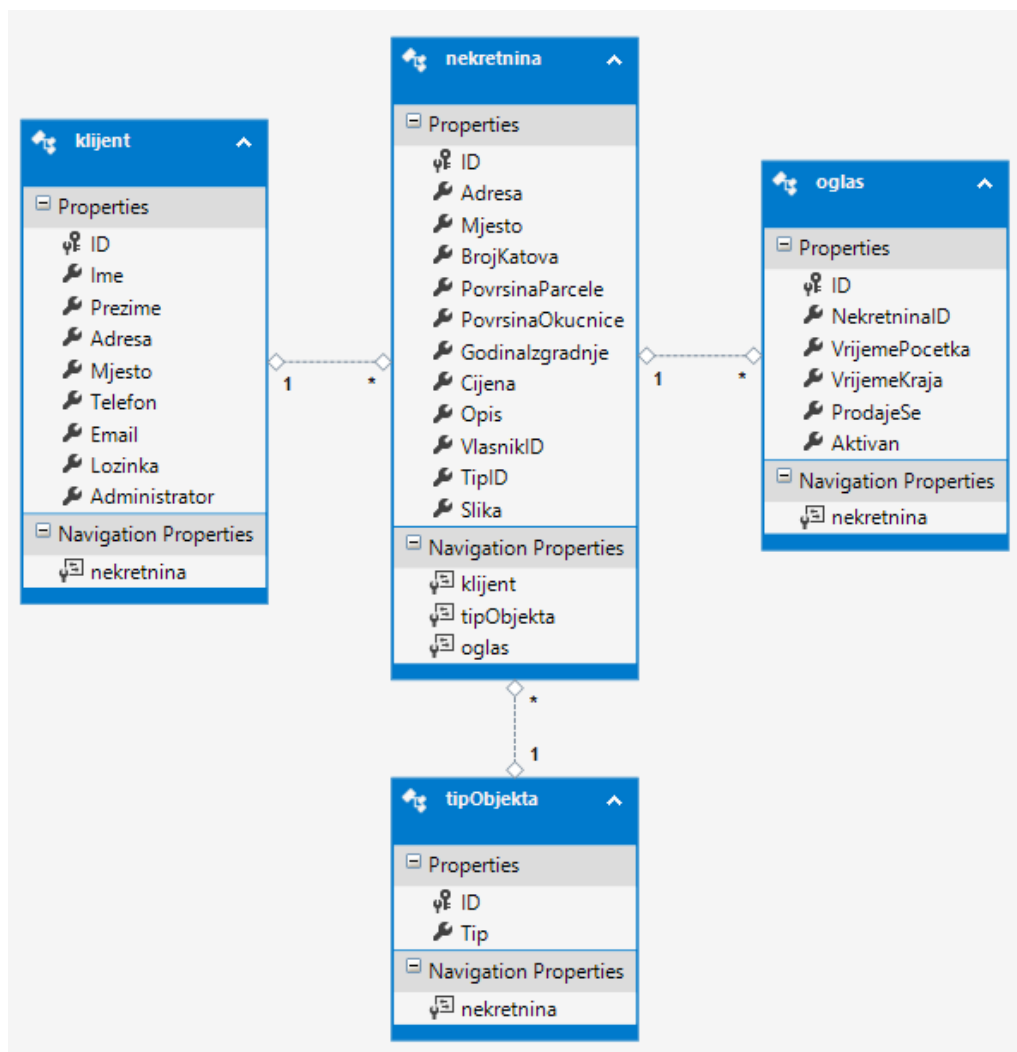
Metoda `OnKeyDown` provjerava da li je na mobilnom uređaju pritisnuta tipka 'back' te ako je vraća `webView` na prijašnju stranicu.

Kod 6. Vraćanje na prijašnju stranicu pomoću tipke „back“

```
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    // provjera da li je pritisnuta tipka 'back'
if((keyCode == KeyEvent.KEYCODE_BACK)) {
if(webview.canGoBack()) {
                // vrati na prijašnju stranicu
webview.goBack();
            } else{
finish();
            }
return true;
}
```

```
returnsuper.onKeyDown(keyCode, event);
}
```

4.3 Baza podatka



Slika 4. Model baze podataka aplikacije za posredovanje nekretninama

Izvor: Vlastiti izvor

4.3.1 Odnos relacija baze podataka

Tablica oglas sadrži polje nekretninaID koji je strani ključ u tablici nekretnina te imaju odnos n:1. Tablica nekretnina sadrži polje vlasnikOIB koji predstavlja strani ključ u relaciji klijent te imaju odnos n:1. Tablica nekretnina također ima polje TipID koji je strani ključ u tablici tipObjekta te ima odnos n:1.

4.3.2 Opis strukture tablica baze podataka

U tablici klijent nalaze se podaci o klijentu i podaci za prijavu te oznaka da li ima administratorske privilegije.

Tablica tipObjekta predstavlja šifrarnik za tipove nekretnina (kuća i stan).

U tablici nekretnina zapisani su podaci o nekretnini te strani ključ na tablicu klijent koji je vlasnik nekretnine te strani ključ na tablicu tipObjekta.

U tablici oglas zapisuje se vrijeme početka i kraja oglasa te oznaka je li je aktivan. Kod predaje oglasa on se postavi kao neaktivan. Oglas može aktivirati samo administrator te se data vrijeme početka oglasa postavi na današnji dan, a kraj oglasa današnji dan uvećan za 30 dana.

4.3.3 Povezivanje baze sa aplikacijom

Model baze podataka je ADO.NET objekt koji se koristi za komunikaciju sa bazom. Napravljen je na principu 'databasefirst'. Prvo je napravljena baza podataka u SQL Serveru te je nakon toga na temelju baze generiran model baze u C# kodu.

Kod 7. prikazuje XML element iz web.config datoteke koji predstavlja connectionstring potreban za spajanje s bazom podataka. Connectionstring unutar sebe sadrži adresu servera, ime baze podataka te autentikacijske podatke.

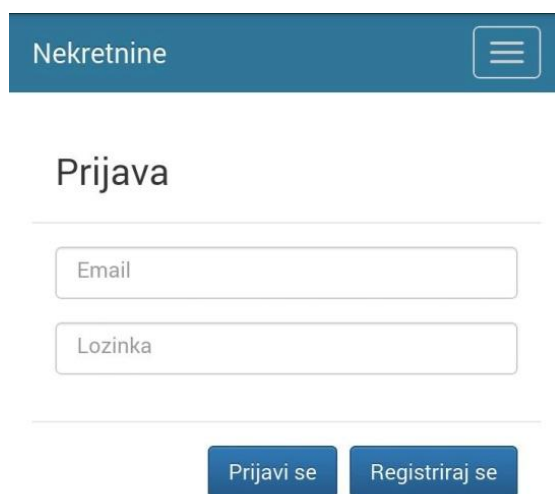
Kod 7.Connectionstringiz web.config-a koji je potreban za komunikaciju s bazom

```
<connectionStrings>
<addname="Entities1"
connectionString="DataSource=mssql5.gear.host;
MultipleActiveResultSets=true;
InitialCatalog=nekretnine;
UserId=nekretnine;
Password=Jj0vE?6rra-h;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

5. Funkcionalnost komponenti aplikacije

5.1 Modul za prijavu korisnika

Modul se koristi za prijavu registriranih korisnika u sustav. Na slici 5. prikazana je forma za prijavu korisnika koja sadrži dva polja za prijavu korisnika. U polje za prijavu korisnici upisuju svoje podatke koji se temelje na e-mailu i lozinci koju su izabrali kod registracije korisnika. Ukoliko korisnik nije izvršio registraciju, pored gumba za prijavu nalazi se gumb za registraciju pomoću kojeg se preusmjeruje na modul za registraciju novog korisnika.



Nekretnine

Prijava

Email

Lozinka

Prijavi se Registriraj se

Slika 5. Prijava korisnika

Izvor: Vlastiti izvor

Metoda `LogIn_Click` unutar web forme `Prijava.aspx.cs` provjerava ispravnost upisanih korisničkih podataka. Kod uspješne autentikacije postavlja se autentikacijski kolačić. Kolačić (eng. Cookie) je podatak koji je pohranjen od strane internet preglednika u svrhe nekih naprednijih funkcionalnosti web stranica. U ovoj aplikaciji koristi se za pohranu informacije o tome da li je korisnik koji trenutno koristi web preglednik prijavljen u aplikaciji. Nakon postavljanja kolačića ovisno o tome da li je korisnik administrator stranice preusmjerava se na administracijski portal ili na stranicu za pretraživanje oglasa. Pri neuspješnoj autentifikaciji aplikacija prikaže odgovarajuću poruku.

Kod 8. Prijava korisnika

```
// provjera da li postoji korisnik s unesenim podacima
if ((klijent = context.klijent.FirstOrDefault(k =>k.Email ==
email &&k.Lozinka == pass)) != null)
{

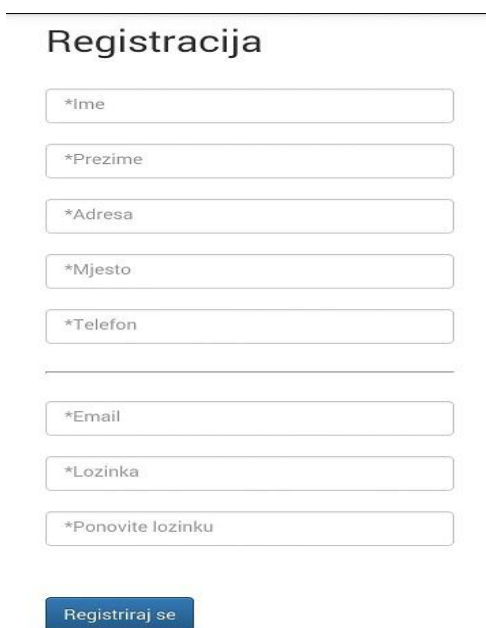
// postavljanje autentikacijskogcookie-a
HttpCookiecookie = new
HttpCookie("ID",klijent.ID.ToString());
Response.SetCookie(cookie);

if (klijent.Administrator)
{
// preusmjeravanje administratora
FormsAuthentication
.SetAuthCookie(klijent.ID.ToString(), true);
Response.Redirect("Admin.aspx");
}
else
{

//preusmjeravanje običnog korisnika
FormsAuthentication
.SetAuthCookie(klijent.ID.ToString(), true);
Response.Redirect("Pretrazivanje.aspx?br=10");
}
}
```


5.2 Modul za registraciju korisnika

Kod registracije korisnika on unosi svoje podatke koji se spremaju u bazu podataka. Kada se podaci o novom korisniku zapišu, korisnik nakon toga može pristupiti prijavi, a potom predaji oglasa. Slika 6. prikazuje polja u koja korisnik unosi svoje podatke, a sa kojim podacima se registrira u aplikaciju te nakon toga sa registriranim podacima može pristupiti prijavi.



The image shows a registration form titled "Registracija". It contains several input fields, each with an asterisk indicating it is required. The fields are: *Ime, *Prezime, *Adresa, *Mjesto, *Telefon, *Email, *Lozinka, and *Ponovite lozinku. Below these fields is a blue button labeled "Registriraj se".

Slika 6. Registracija korisnika

Izvor: vlastiti izvor

Kod 9. prikazuje odsječak metode RegisterUser_Click koja se nalazi unutar web forme Registracija.aspx.cs. U odsječku se provjerava postoji li u bazi korisnik s unesenim e-mailom. Ukoliko ne postoji te se lozinka i potvrda lozinke ne podudaraju, on se dodaje u bazu kao novoregistrirani korisnik. Nakon toga aplikacija se preusmjerava na web formu UspješnaRegistracija.aspx. Ako registracija nije bila uspješna, prikazuje se odgovarajuća poruka.

Kod 9. Provjera postojećeg ili novog korisnika

```
// provjera da li već postoji korisnik s unešenim email-om
if (!context.klijent.Any(klijent =>klijent.Email == k.Email))
{
    // provjera da li je lozinka i ponovljena lozinka ista
    if (k.Lozinka == passwordRepeat)
    {
        context.klijent.Add(k);
        if (context.SaveChanges() > 0)
        {
            //preusmjeravanje pri uspješnoj registraciji
            Response
            .Redirect("UspješnaRegistracija.aspx?id=success");
        }
    }
    else
    {
        Label1.Text = "Neuspješna registracija.";
        Label1.CssClass = "bg-danger";
        Label1.Visible = true;
    }
}
else
{
    Label1.Text = "Greška: Lozinke se ne podudaraju.";
    Label1.CssClass = "bg-danger";
    Label1.Visible = true;
}
else
{
    Label1.Text = "Greška: Uneseni email već postoji.";
    Label1.CssClass = "bg-danger";
    Label1.Visible = true;
}
```

Na web formi UspješnaRegistracija.aspx.cs unutar metodePage_Load provjerava se vrijednost parametra 'id' koji se šalje preko URL-a. Ako mu je vrijednost 'success', korisniku se prikazuje poruka o uspješnosti registracije. Nakon par sekundi korisnik će automatski biti preusmjeren na web formu Prijava.aspx.

Kod 10. Javascript kod koji nakon 3 sekunde preusmjerava korisnika na web formu
Prijava.aspx

```
<script type="text/javascript">
    (function () {
    setTimeout(function () {
    window.location = "/Prijava.aspx";
        }, 3000); /* 1000 = 1 second*/
    }) ();
</script>
```

5.3 Administracija podataka i korisnika

Forma administratora prikazana na slici 7. sadrži tablicu sa podacima svih oglasa, objavljenih i neobjavljenih. Forma administrator sadrži opcije aktivacije i deaktivacije oglasa, kao i brisanje istih. Nakon prijave administratora aplikacija se automatski preusmjerava na web formu za upravljanje oglasima.

Nekretnine

Pretraživanje

Predaj oglas

Registracija

Prijava

Dino

Odjava

Oglasi

ID	Adresa	Mjesto	Cijena [kn]	Tip	Površina [m2]	Površina Okućnice [m2]	Broj Etaža	Objavljeno	Ističe	Aktivan			
1	Trg bana Josipa Jelacica 5	Zagreb	350000	Stan	60	0	2	07/06/2016	07/07/2016	<input checked="" type="checkbox"/>	Aktiviraj	Deaktiviraj	Izbrisi
2	Krizevacka ulica 24	Koprivnica	400000	Kuca	110	100	1	07/06/2016	07/07/2016	<input checked="" type="checkbox"/>	Aktiviraj	Deaktiviraj	Izbrisi
3	Ulica grada Vukovara	Zagreb	500000	Stan	65	0	2	07/06/2016	07/07/2016	<input checked="" type="checkbox"/>	Aktiviraj	Deaktiviraj	Izbrisi
4	Ulica Matice hrvatske 55	Velika Gorica	400000	Kuca	130	100	1	07/06/2016	07/07/2016	<input checked="" type="checkbox"/>	Aktiviraj	Deaktiviraj	Izbrisi
5	V. Nazora 30	Čakovec	204000	Stan	80	0	1	17/06/2016	17/07/2016	<input checked="" type="checkbox"/>	Aktiviraj	Deaktiviraj	Izbrisi

Slika 7. Administracija oglasa

Izvor: vlastiti izvor

Kod učitavanja web forme Admin.aspx poziva se funkcija GetOglasi koja dohvaća sve oglase iz baze.

Kod 11. Dohvaćanje oglasa iz baze

```
// spajanje relacije oglas i relacije nekretnina
var oglasNekretnina = context.oglas.Join(
    context.nekretnina,
    o => o.NekretninaID,
    n => n.ID,
    (o, n) => new { O = o, N = n })
    .ToList(); // pretvaranje u listu
```

Kod 12. prikazuje metodu BuildColumn skoja definira stupce za objekt DataTable gdje će se dohvatiti oglasi iz baze.

Kod 12. Metoda BuildColumns

```
private DataTable BuildColumns(DataTable dt)
{
    dt.Columns.Add("Objava", typeof(string));
    dt.Columns.Add("Mjesto", typeof(string));
    dt.Columns.Add("Tip", typeof(string));
    dt.Columns.Add("Vrsta", typeof(string));
    dt.Columns.Add("Cijena", typeof(string));
    dt.Columns.Add("NekretninaID", typeof(int));
    return dt;
}
```

Kod 13. prikazuje odsječak koda DataTable objekt koji sadrži oglase dohvaćene iz baze, postavlja kao izvor podataka GridView tablicu.

Kod 13. Postavljanje izvora podataka za GridView

```
if (dt.Rows.Count > 0)
{
    // postavljanje izvora podataka za GridView
    GridView1.DataSource = dt;
    GridView1.DataBind();
    GridView1.Visible = true;
}
```

Klikom na neku naredbu u nekom retku GridView tablice poziva se funkcija GridView1_RowCommand. Na temelju ID-a retka, odabrani oglas se dohvaća iz baze te ovisno o naredbi koja je odabrana, izvršava se aktivacija, deaktivacija ili brisanje oglasa.

Kod 14. prikazuje dohvaćanje oglasa iz baze na temelju ID stupca pritisnutog retka te izvršavanje metoda aktivacije, deaktivacije i brisanja oglasa.

Kod 14. Upravljanje oglasima

```
ProtectedvoidGridView1_RowCommand(objectsender,
GridViewCommandEventArgs e)
{
    // dohvaćanje pritisnutog retka
    intindex = Convert.ToInt32(e.CommandArgument);
    GridViewRowrow = GridView1.Rows[index];
    intoglasID = Convert.ToInt32(row.Cells[0].Text);
    // dohvatanje oglasa na temelju ID-a
    var oglas = context.oglas
.Where(x => x.ID == oglasID).FirstOrDefault();
    if (e.CommandName == "activate")
    {
        if (!oglas.Aktivan)
        {
            //aktivacija oglasa ako već nije aktivan
            oglas.Aktivan = true;
            oglas.VrijemePocetka = DateTime.Now;
            oglas.VrijemeKraja = DateTime.Now
            .Add(newTimeSpan(30, 0, 0, 0));
            context.SaveChanges();
            // osvježavanje GridView-a
            GetOglasi();
        }
    }
    if (e.CommandName == "deactivate")
    {
        if (oglas.Aktivan)
        {
            // deaktivacija oglasa ako je aktivan
            oglas.Aktivan = false;
            context.SaveChanges();

            // osvježavanje GridView-a
            GetOglasi();
        }
    }
    if (e.CommandName == "erase")
```

```
{
    // brisanje oglasa
    context.oglas.Remove(oglas);
    context.SaveChanges();

    // osvježavanje GridView-a
    GetOglasi();
}
}
```

5.4 Modul za predaju oglasa

Da bi se izvršila predaja oglasa korisnik mora biti registriran i prijavljen u aplikaciju. Forma za predaju oglasa prikazana je na slici 8. i sastoji se od polja za upis adrese, mjesta u kojem se nekretnina nalazi, površine nekretnine, površine okućnice, godine izgradnje, cijene. Korisnik može odabrati sliku, tip nekretnine i broj etaža nekretnine. Može se također dodati opis ako korisnik koji stavlja nekretninu na prodaju želi opisati istu svojim riječima te tako obogatiti oglas. Nakon upisanih podataka korisnik pritiskom na gumb „pošalji“, oglas zapisuje u bazu podataka te zaprimljena prijava treba biti odobrena od strane administratora. Nakon što administrator prihvati oglas i odobri isti, oglas je vidljiv 30 dana te se nakon toga može ponovno odobriti.

Predaj oglas

Tip:

*Adresa

*Mjesto

Broj Etaža:

*Površina [m2]

*Površina Okućnice [m2]

*Godina Izgradnje

*Cijena [kn]

Odaberite sliku

Opis

Slika 8. Forma za upis podatka novog oglasa

Izvor: vlastiti izvor

Kod 15. prikazuje odsječak metode `PredajOglas_Click` koja se nalazi unutar web forme `PredajOglas.aspx.cs`. U odsječku koda se iz vrijednosti unesenih u formi stvori novi objekt razreda `nekretnina` te ga se zapisuje u bazu. Nakon toga se stvori novi objekt razreda `oglas` te ga se povezuje s nekretninom preko njezinog ID-a (`NekretninaID`).

Kod 15. Odsječak metode `PredajOglas_Click` koji sprema nekretninu i oglas u bazu

```
// dodavanje novostvorene nekretnine u bazu
context.nekretnina.Add(nek);
if (context.SaveChanges() > 0)
{
    // stvaranje oglasa te povezivanje s nekretnine preko ID-a
    oglas og = new oglas();
    og.NekretninaID = nek.ID;
    og.VrijemePocetka = DateTime.Now;
```

```
og.VrijemeKraja =  
DateTime.Now.Add(new TimeSpan(30, 0, 0, 0));  
og.Aktivan = false;  
  
    // dodavanje oglasa u bazu  
context.oglas.Add(og);  
if (context.SaveChanges() > 0)  
{  
    // prikaz poruke o uspješnosti  
Label1.Text = "Oglas je uspješno predan na odobrenje.";  
Label1.CssClass = "bg-success";  
Label1.Visible = true;  
return;  
}  
}
```

5.5 Modul za pretraživanje oglasa

Pretraživanje oglasa odvija se na temelju upisanih parametara u filter formi što je prikazano na slici 9. Korisnik za filtriranje može upisati samo određeni parametar npr. cijenu te mu aplikacija na temelju unosa željene cijene izbaciti nekretnine u tom cjenovnom okviru, bez obzira jesu li uneseni ostali parametri u filter za pretraživanje. Također, korisniku se nudi opcija pretraživanja svih oglasa bez unošenja parametara u filter za pretraživanje. Pretraživanje oglasa sastoji se od polja u kojem se odabire vrsta oglasa, polja u kojem se upisuje mjesto gdje se nalazi nekretnina, polje za vrstu objekta, polje za cjenovni okvir izražen u kunama te polja za unos površine. Kada korisnik pritisne gumb pretraži oglase, aplikacija filtrira podatke iz baze podataka, prikaže prozor sa podacima koji zadovoljavaju traženi kriterij što je vidljivo na slici 10. koja prikazuje osnovne parametre koje je korisnik unio u filter pretraživanja te na temelju vlastitog izbora može odabrati pojedini oglas koji se prikaže u skočnom prozoru, a koji nudi više detaljnih podataka o nekretnini prikazanih na slici 11. sa Google kartom lokacije nekretnine te ako je korisnik koji je predao oglas stavio sliku u oglas, dobije se prikazani oglas sa slikom.

Nekretnine Dino Odjava

Pretraživanje

Vrsta oglasa:

Mjesto

Vrsta objekta:

Cijena [kn]

Od: Do:

Površina [m²]

Od: Do:

Slika 9. Forma za pretraživanje oglasa

Izvor: vlastiti izvor

Pretraživanje

Objava	Mjesto	Vrsta objekta	Vrsta oglasa	Cijena	
07.06.2016	Trg bana Josipa Jelacica 5, Zagreb	Stan	Prodaja	350000 kn	→
07.06.2016	Krizevacka ulica 24, Koprivnica	Kuca	Prodaja	400000 kn	→
07.06.2016	Ulica grada Vukovara, Zagreb	Stan	Prodaja	500000 kn	→
07.06.2016	Ulica Matice hrvatske 55, Velika Gorica	Kuca	Prodaja	400000 kn	→
17.06.2016	V. Nazora 30, Čakovec	Stan	Kupnja	204000 kn	→
17.06.2016	nazora 30, čakovec	Stan	Kupnja	100200 kn	→
19.06.2016	Hrvatske Državnosti , Koprivnica	Kuca	Kupnja	500000 kn	→

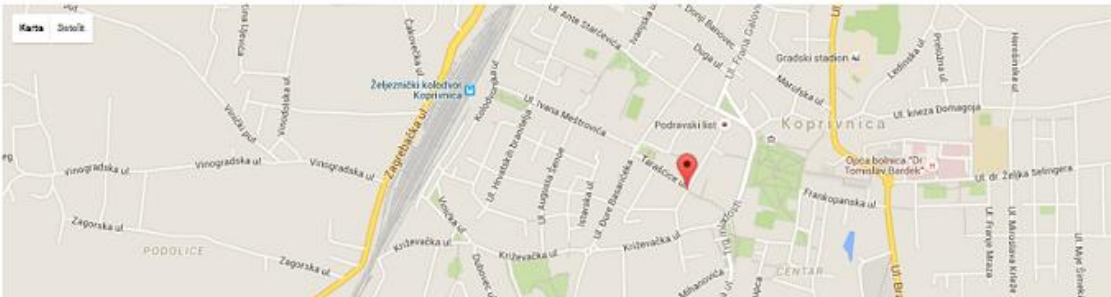

Slika 10. Prikaz rezultata pretraživanja oglasa nakon upisanih podataka kao rezultat odabira podataka.

Izvor: vlastiti izvor

Podaci o nekretnini

[← povratak na rezultate](#)

Adresa	Hrvatske Državnosti
Mjesto	Koprivnica
Broj Etaža	1
Površina Parcele	80 m ²
Površina Okućnice	300 m ²
Godina Izgradnje	2013
Cijena	500000 kn
Vrsta oglasa	Kupnja
Opis	Novogradnja



Slika 11. Prikaza detalja odabranog oglasa

Izvor: vlastiti izvor

Pretrazivanje.aspx sadrži polja za pretragu oglasa, GridView tablicu koja je na početku skrivena te nakon pokrenute pretrage tablica postane vidljiva. Na stranici se nalazi skriveni Bootstrap modal popup koji sadrži polja za podatke o pojedinoj nekretnini i kartu koja prikazuje adresu. Polja se popune dok se izvrši odabir na određeni oglas u tablici. Pozicija na karti postavlja se pomoću Googleove skripte. Koristi se i javascript funkcija koja šalje AJAX poziv AutoCompleteService.aspx servisu na svaki klik unutar Mjesto polja.

AutoCompleteService.aspx sadrži metodu GetMjesta koja od Pretrazivanje.aspx stranice ajax pozivom prima parametar 'prefix'.

Kod 16. prikazuje metodu GetMjesta koja prima parametar tipa string, a kao rezultat vraća sve nekretnine kojima atribut mjesto započinje s parametrom prefix. Iz web.configa se dohvaća connectionstring pomoću kojeg se stvori SqlConnection objekt koji služi za komunikaciju s bazom. Zatim se stvara objekt SqlCommand te se kao vrsta

sqlnaredbe postavlja pohranjena procedura definirana u datoteci Nekretnine.sql. Objektu SqlCommand se kao parametar postavlja string koji je metoda GetMjesta primila. Izvršava se upit te se rezultati spremaju u listu stringova.

Kod 16. Pretraživanje mjesta po prefixu.

```
public List<string>GetMjesta(stringprefix)
{
    List<string> mjesta = new List<string>();
    // dohvat connectionstringa iz web.configa
    stringcs = ConfigurationManager
        .ConnectionStrings["Entities1"].ConnectionString;
    // definiranje konekcije prema bazi
    using (SqlConnectioncon = newSqlConnection(cs))
    {
        // definiranje naredbe
        SqlCommandcmd =
            newSqlCommand("MjestaKojaPocinjuSa", con);
        cmd.CommandType = CommandType.StoredProcedure;
        // definiranje parametra
        SqlParameterparameter =
            newSqlParameter("@prefix", prefix);
        cmd.Parameters.Add(parameter);

        // otvaranje konekcije
        con.Open();

        // izvorsavanje naredbe
        SqlDataReader rdr = cmd.ExecuteReader();

        // učitavanje rezultata u listu
        while (rdr.Read())
        {
            mjesta.Add(rdr["Mjesto"].ToString());
        }
    }
    return mjesta;
}
```

Unutar metodePage_Load na web formi Pretraživanje.aspx.cs provjerava se ima li parametar 'br' ima vrijednost 10. Ako parametar 'br' ima vrijednost 10 onda se prikazuje prvih 10 oglasa iz baze jer se taj parametar šalje samo nakon uspješne prijave. Također, poziva se i metodaGetMjesta koja ne prima parametar te iz baze dohvaća mjesta svih nekretnina koje se nalaze u bazi.

Metoda SviOglasi_Click prvo dohvaća oglase koju su istekli te ih deaktivira ako su još uvijek aktivni.

Kod 17. Odječak metoda SviOglasi_Click koji kontrolira istekle oglase

```
var istekliOglasi = context.oglas.Where(  
o =>o.VrijemeKraja<DateTime.Now&&o.Aktivan).ToList();  
foreach (var ntorkainistekliOglasi)  
{  
    // deaktivacija oglasa koji su istekli  
    ntorka.Aktivan = false;  
    context.SaveChanges();  
}
```

Zatim se dohvaćaju svi aktivni oglasi te se dodaju u objekt DataTable koji se postavi kao izvor podataka za GridView.

Metoda Pretraga_Click prvo deaktivira istekle oglase, zatim dohvati sve oglase koji zadovoljavaju min i max vrijednosti cijene i površine parcele te oznaku prodaje li se ili kupuje. Zatim se još ti oglasi filtriraju ovisno o tome je li unesen podatak u polje mjesto i je li odabran tip nekretnine. Preostali oglasi se dodaju u DataTable objekt koji se postavi kao izvor podataka za GridView.

Kod 18. Odsječak metode Pretraga_Click koja filtrira rezultate prema unesenim parametrima

```
var oglasNekretnina =  
context.oglas.Where(x =>x.Aktivan)  
    // spajanje relacije oglas i relacije nekretnina  
.Join(  
context.nekretnina,  
o =>o.NekretninaID,  
    n => n.ID,  
(o, n) =>new { O = o, N = n })  
    // filtriranje po cijeni, površini i vrsti oglasa  
.Where(  
a =>a.O.ProdajeSe == prodajeSe&&  
a.N.Cijena>= minCijena&&
```

```

a.N.Cijena<= maxCijena&&
a.N.PovrsinaParcele>= minPov&&
a.N.PovrsinaParcele<= maxPov).ToList();
if (mjesto != "")
{
    oglasNekretnina = oglasNekretnina
        // filtriranje po mjestu nekretnine
    .Where(x =>x.N.Mjesto.ToLower()
    == mjesto.ToLower()).ToList();
}
if (tip != -1)
{
    // filtriranje po tipu nekretnine
    oglasNekretnina = oglasNekretnina
    .Where(x =>x.N.TipID == tip).ToList();
}

```

Metoda `GridView1_Sorting` pozove se kad se klikne na ime stupca. Metoda sortira oglase po tom stupcu po abecedi uzlazno ili silazno (engl. ascending, descending). U Session varijablu pomoću metode `ConvertSortDirection` postavlja se je li zadnje sortiranje bilo uzlazno ili silazno kako bi se dobilo da sortiranje alternira. Na prvi klik silazno, drugi klik uzlazno, treći opet silazno itd.

Kod 19. Sortiranje oglasa

```

protectedvoidGridView1_Sorting(objectsender,
GridViewSortEventArgs e)
{
    DataTabledataTable = ViewState["dtbl"] as DataTable;
    if (dataTable != null)
    {
        DataViewdataView = newDataView(dataTable);
        if (Session["lastSort"] != null)
        {
            // odlucivanje smjera sortiranja ovisno o trenutom
            stringlastSort = Session["lastSort"].ToString();
            stringsortDirection =
            (lastSort == "ASC") ? "DESC" : "ASC";
            //sortiranje podataka prema kliknutom stupcu i smjeru
            dataView.Sort = e.SortExpression+" "+sortDirection;
            SortDirectionsd =
            sortDirection == "ASC" ?

```

```
SortDirection.Ascending :
SortDirection.Descending;
Session["lastSort"] = ConvertSortDirection(sd);
}
else
{
    dataView.Sort = e.SortExpression + " " +
    ConvertSortDirection(e.SortDirection);
    Session["lastSort"] =
    ConvertSortDirection(e.SortDirection);
}
    // osvježavanje podataka u GridView-u
GridView1.DataSource = dataView;
GridView1.DataBind();
}
}
```

6. Konfiguriranje aplikacija

U web.config-ima na kraju datoteke nalaze se connection String-ovi za komunikaciju s bazom. Za potrebe ovog projekta korištena su dva connectionString-a. Prvi koristi ADO model pri uobičajenoj komunikaciji, a drugi AutoComplete servis kada poziva pohranjenu proceduru. Kod povezanosti opisan je u podnaslovu baza podataka. web.config također sadrži i konfiguraciju načina autentikacije. Web aplikacija koristi FormsAuthentication način autentikacije koji koristi Http cookie kako bi čuvao informaciju o korisnikovoj prijavi.

Kod 20. prikazuje odsječak iz web.config datoteke koji definira način autentikacije koji aplikacija koristi. Ovdje se definira loginUrl na koji aplikacija preusmjerava u slučaju da korisnik koji nije registriran pokuša pristupiti nekoj stranici koja zahtjeva da korisnik bude autentificiran.

Kod 20. Konfiguracija načina autentikacije

```
<authentication mode="Forms">
<forms
loginUrl="Prijava.aspx"
defaultUrl="Pretrazivanje.aspx"
```

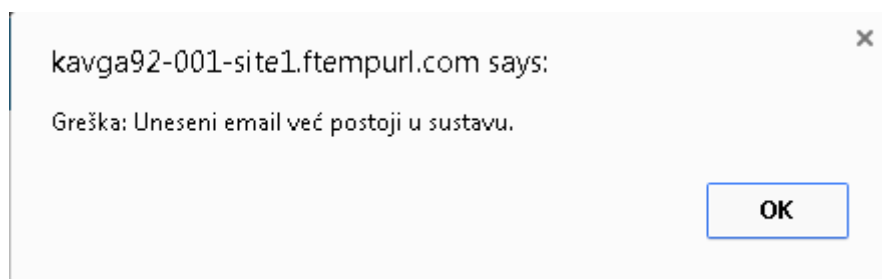
```
name=".ASPXFORMSAUTH">  
</forms>  
</authentication>
```

Za pohranu slike koristi se razred `WebChromeClient` kako bi omogućili učitavanje slike. Nakon toga pokreće se `Intent` za odabir slike sa lokalne pohrane sa mobilnog uređaja. Slika se sprema u bazu podataka u kojoj je jedno polje rezervirano za sliku. Slika se sprema u formatu `jpg` kodirana u `base64` formatu. Kod za pohranu slike može se vidjeti pod naslovom struktura mobilne aplikacije. Povezivanje mobilne aplikacije za web i njezino konfiguriranje opisano je u naslovu struktura aplikacije, u podnaslovu struktura mobilne aplikacije, a detaljno je opisana u kodu 4.

7. Testiranje aplikacije

Tijekom razvoja aplikacije provedena su testiranja njene funkcionalnosti, provjera registracije istog korisnika, unos podatka i aktivacija oglasa.

Kod početnog testiranja aplikacije, krenuo sam sa testiranjem registracije korisnika pokušao sam upisati iste podatke, ako su podaci već zapisani u bazu javlja se poruka koja je prikazana na slici 12.



Slika 12. Registracija istog korisnika

Izvor: vlastiti izvor

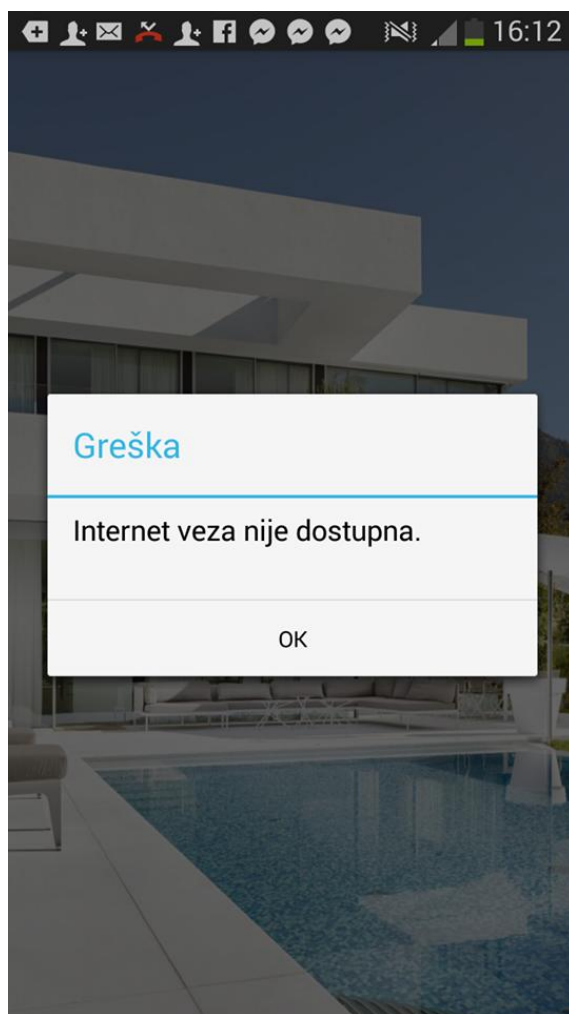
Ako korisnik želi pristupiti predaji oglasa mora izvršiti prijavu. Ukoliko se korisnik nije prijavio, a želi pristupiti modulu za predaju oglasa, aplikacija ga automatski prebacuje na modul za prijavu korisnika. To je izvršeno sa ciljem da se korisniku pruži jednostavnost i funkcionalnost ove aplikacije što mi je kao autoru bio i cilj. Korisniku koji se prvi put susreće sa ovakvom aplikacijom da ne budu „zbunjen“, nego da ga aplikacija „sama vodi“ do željenog cilja, a to je predaja oglasa ili pregled.

Testiranje pretraživanje oglasa - testiranje je provedeno na način da se testirala brzina pregleda željenih oglasa te kvaliteta željenih informacija. U polje su se unosili parametri, npr. pretraga oglasa prema parametru mjesta ili vrsti objekta, aplikacija je vratila podatke koji su bili zatraženi prema unesenim parametrima. To znači da nije bilo potrebno ispuniti sva polja da bi se prikazali oglasi, nego je to učinjeno samo prema željenom parametru, a ostali podaci nad kojima nije bio zadan uvjet bili su prikazani. Također pokušano je pretraživati oglase na temelju unesenog prefixa mjesto, znači da su u bazu zapisana sva mjesta te se odmah prikažu sva mjesta na temelju unesenog prefixa npr. ako korisnik napiše prefix zag aplikacija mora ponuditi Zagreb, aplikacija nije osjetljiva na mala i velika slova. Time je omogućen brži unos podataka što općenito znači da aplikacija brže radi. Detaljnije o pretraživanju može se vidjeti u naslovu funkcionalnost komponenti aplikacije u podnaslovu modul za pretraživanje oglasa.

Testiranje predaje oglasa, testiranje je bilo provedeno na način da se testirala pouzdanost informacija koje se daju korisnicima na pregled. Primjer, trenutno je popularno da na sličnim aplikacijama registrirani korisnici predaju neke lažne oglase sa primamljivim cijenama te na taj način privuku „naivne“ korisnike koji pregledavaju oglase na prevaru. U takvim aplikacijama nije izvršena provjera sigurnosti i pouzdanosti informacija oglasa od strane administratora te administratori onda naknadno nakon što je oglas predan pregledavaju i brišu sumnjive oglase. U ovoj aplikaciji to je osigurano na način da kada korisnik preda oglas, oglas ne može biti vidljiv korisnicima dok administrator ne izvrši provjeru oglasa te prema vlastitoj procjeni utvrdi hoće li odobriti oglas za pregled ili ne. Itestirano je i slanje slike na server koja je velikog formata npr. 1280x1024, a aplikacija je automatski prilagodila veličinu slike stranici.

Pokušano je testirati što se dogodi ako mobilni uređaj izgubi internet vezu. Prikazan

je prozor upozorenja za obavijest o izgubljenju internet vezi što se može vidjeti na slici 13. Slika prikazuje obavijest koja upozorava korisnika zašto povezivanje s aplikacijom nije moguće jer kod drugih raznih aplikacija ako nema ovakvog upozorenja odmah korisnici sumnjaju da je problem u samoj aplikaciji, a ne u izgubljenju internet vezi. Tako korisnik odmah može eliminirati problem.



Slika 13. Poruka o izgubljenju internet vezi

Izvor: Vlastiti izvor

8. Zaključak

Ovim završnim radom prikazan je razvoj aplikacije za posredovanje nekretninama u kojoj razvijena aplikacija omogućuje da korisnik ima uvid u predane oglase te pregled istih bez potrebne prijave. Detaljno su opisani moduli aplikacije i njihova povezanost. Prilikom izrade aplikacije korištene su tehnologije zasnovane na ASP.NET i android platformi. Upotrebom ove aplikacije korisnik može pregledavati oglase na svom mobilnom uređaju bilo gdje, samo korisnikov uređaj mora ostvariti internetsku vezu. Korisniku se omogućuje pregled oglasa sa slikom i lokacijom nekretnine koja je vidljiva na Google karti u aplikaciji. Najteži dio izrade praktičnog djela rada bio je uskladiti web aplikaciju s mobilnom aplikacijom. Unatoč web-view elementu, potrebno je bilo uskladiti određene komponente u androidu kako bi mobilna aplikacija radila kao i web aplikacija. Od aplikacije se tražilo da podržava određenu razinu sigurnosti podataka korisnicima jer administrator odlučuje o eliminaciji sumnjivih oglasa te se korisnici mogu pouzdati u točnost podataka - to je bila ideja prije izrade ovog projekta i u potpunosti je realizirana. Ova aplikacija razvijena je tako da pruža jednostavnost usluge i istu mogu koristiti osobe koje imaju osnovna znanja u korištenju interneta i tehnologija. Aplikaciju je moguće nadograditi s dodatnim modulima koji će omogućiti da se kontakt sa korisnikom koji je predao oglas može uspostaviti putem prozora za komunikaciju koji bi se otvorio kod pregleda oglasa, tako bi ova aplikacija bila još kompleksnija.

Literatura

Internet izvori:

1. IDC statistički podaci (12.6.2016.)
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
2. Android operacijski sustav (24.6.2017.)
https://wiki.open.hr/wiki/Android_OS
3. Android operacijski sustav (21.6.2017.)
[https://hr.wikipedia.org/wiki/Android_\(operacijski_sustav\)](https://hr.wikipedia.org/wiki/Android_(operacijski_sustav))
4. Arhitektura android sustava (21.6.2017.)
<https://developer.android.com/guide/platform/index.html>(19.6.2016.)
5. Ivica Glavaš, Dizajn WEB korisničkih sučelja, ASP.NET tehnologija, 2003.
<http://web.zpr.fer.hr/ergonomija/2003/glavas> (23.2.2015.)
6. Bootstrap(24.6.2017.)
<http://whatis.techtarget.com/definition/bootstrap>
7. Bootstrap(1.4.2016.)
<http://blog.falcon-tech.rs/2015/04/22/sta-je-bootstrap/sta-je-bootstrap/>
8. Entity Framework (12.6.2016.)
<https://msdn.microsoft.com>
9. CodyLindley Osnove biblioteke jQuery(25.8.2016.)
http://www.mikroknjiga.rs/Knjige/JQK/01_JQK.pdf